

Happy 😂 Version Control for your Power BI Dashboards 📊

made with ❤️ by Adriel E. González Martínez for eIQS Puerto Rico

The problem

1. Power BI files (**.pbix**) are binary, meaning we can not track changes at the content level like we do it for text-based files (e.g., .sql, .json, .txt).
2. You can't easily view diffs between different versions of **.pbix** files.

The (or at least a possible) Solution

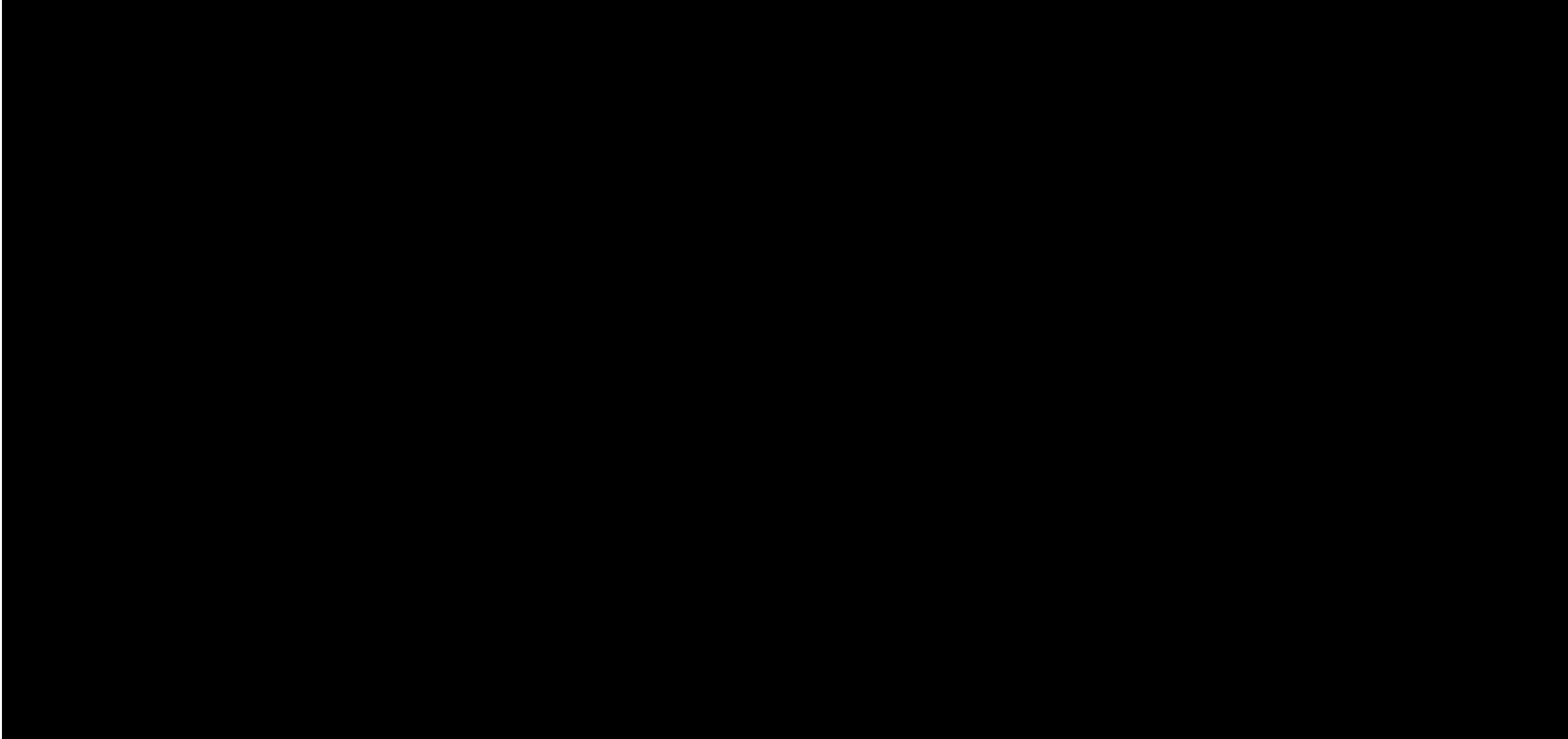
1. Close your eyes 🙄 and hope for the best
2. Or, Save their work in **.PBIP** file format, which breaks down into **JSON** components.
This format makes the files editable in tools like VS Code.

STEP #1: Let's set up your Environment

What tools you will need?

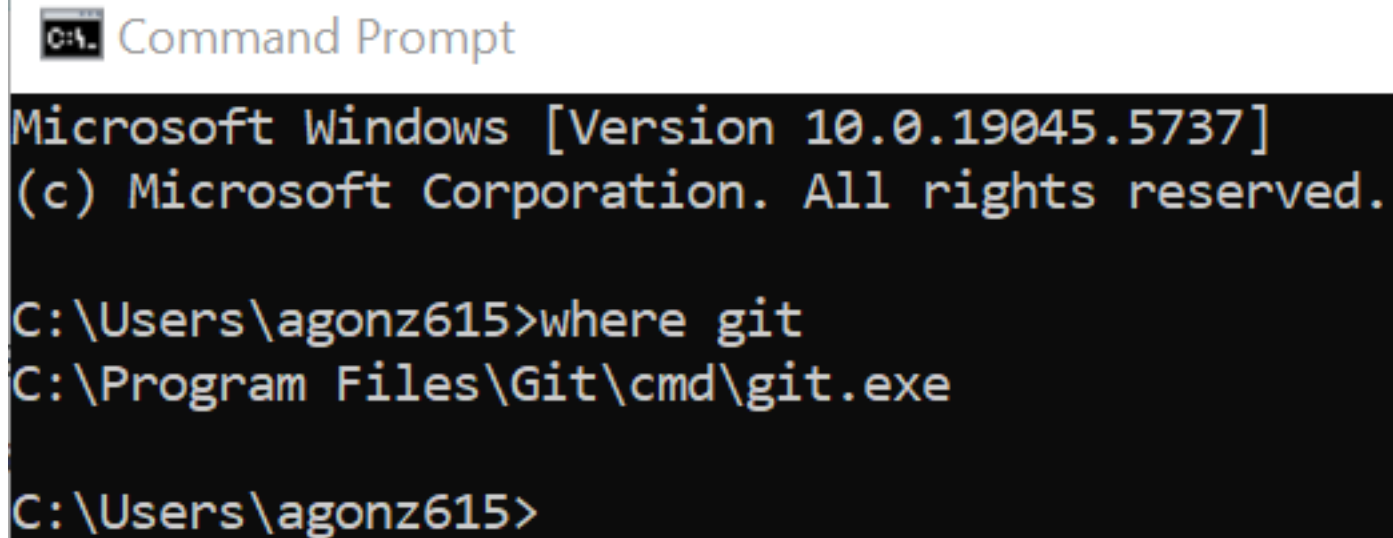
1. [Power BI Desktop](#)
2. [Visual Studio Code](#)
3. [Git](#)
4. [Github Account](#)

How to install Git (The UHG way...)



IF Git truly installed?

```
C:\Users\agonz615>where git
```



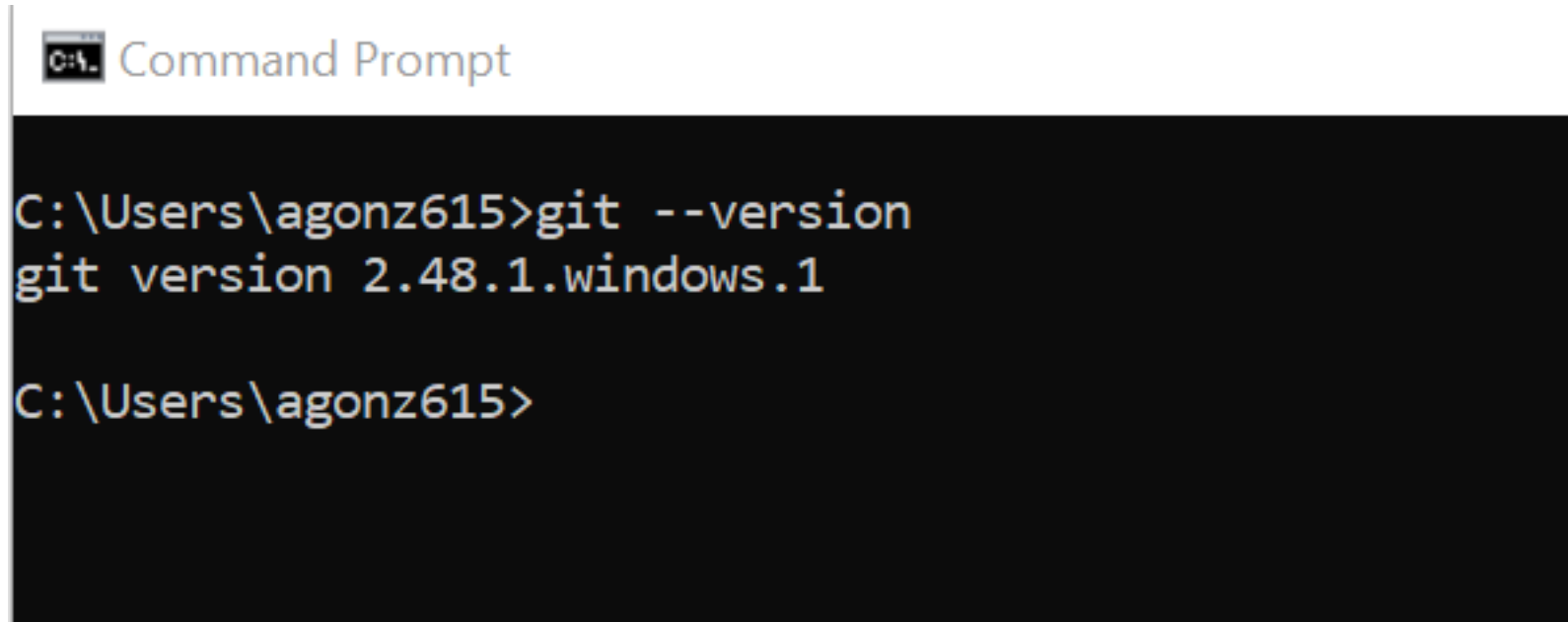
The screenshot shows a Windows Command Prompt window with the title bar 'C:\ Command Prompt'. The window content displays the following text: 'Microsoft Windows [Version 10.0.19045.5737]', '(c) Microsoft Corporation. All rights reserved.', 'C:\Users\agonz615>where git', 'C:\Program Files\Git\cmd\git.exe', and 'C:\Users\agonz615>'.

```
C:\ Command Prompt  
Microsoft Windows [Version 10.0.19045.5737]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\agonz615>where git  
C:\Program Files\Git\cmd\git.exe  
  
C:\Users\agonz615>
```

A (Windows) command prompt showing the command to locate the path to a given file.

But, What version am I running?

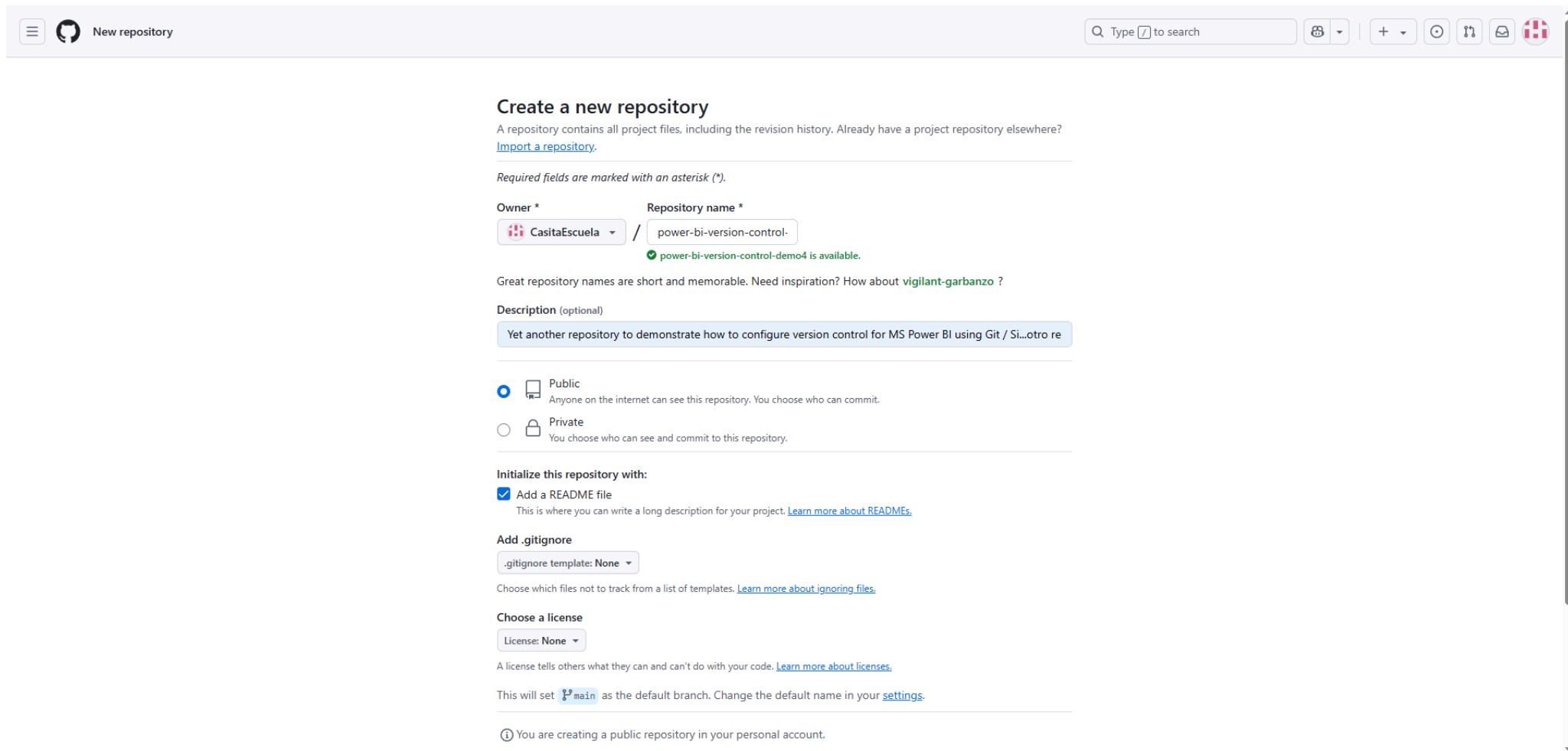
```
C:\Users\agonz615>git --version
```

A screenshot of a Windows Command Prompt window. The title bar at the top reads "Command Prompt". The command prompt shows the user's current directory as "C:\Users\agonz615". The user has entered the command "git --version", and the output displayed is "git version 2.48.1.windows.1". The prompt is ready for the next command.

```
C:\Users\agonz615>git --version  
git version 2.48.1.windows.1  
  
C:\Users\agonz615>
```

A (Windows) command prompt showing the git command with the version parameters.

STEP #2: Create a (REPO)sitory



New repository

Q Type to search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

CasitaEscuela / power-bi-version-control-

power-bi-version-control-demo4 is available.

Great repository names are short and memorable. Need inspiration? How about [vigilant-garbanzo](#) ?

Description (optional)

Yet another repository to demonstrate how to configure version control for MS Power BI using Git / Si...otro re

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

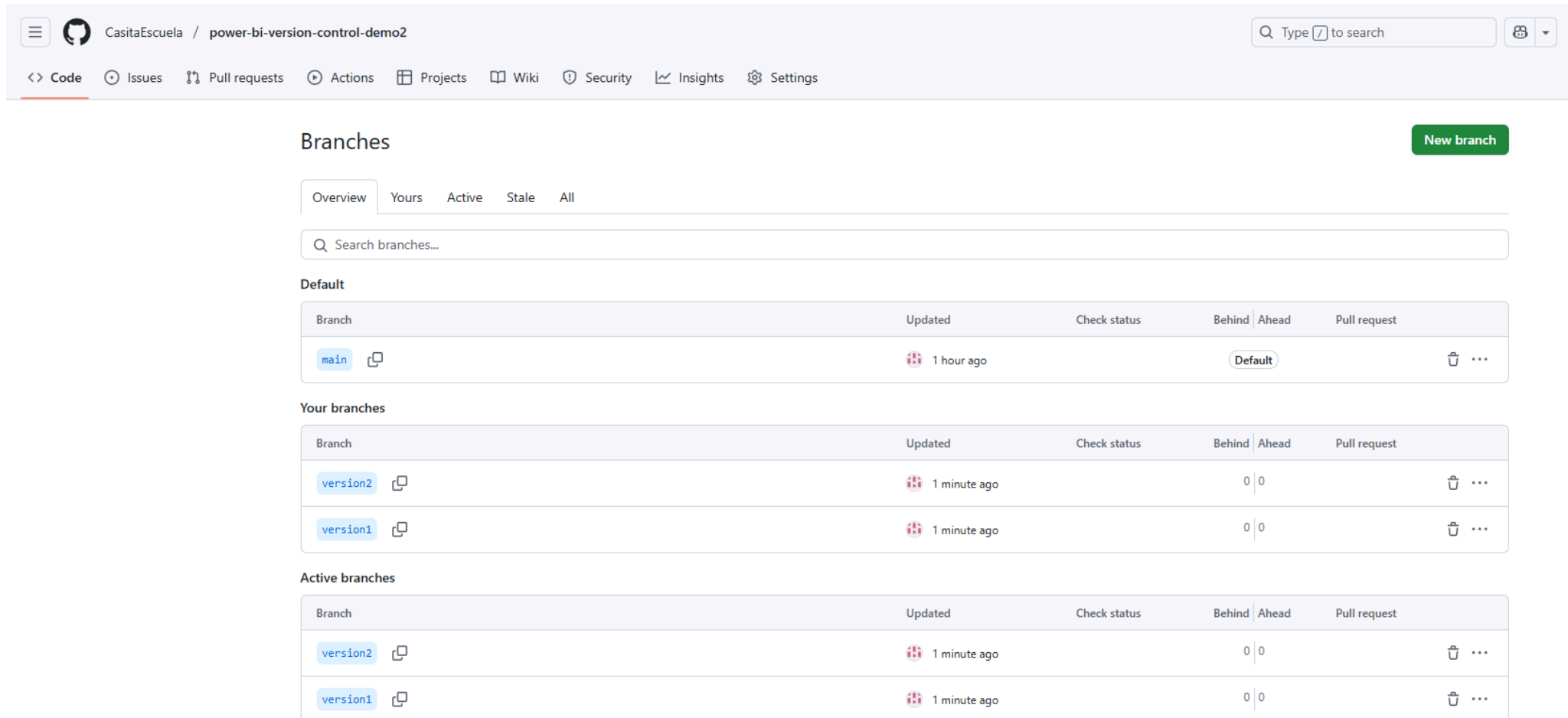
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

📘 You are creating a public repository in your personal account.

A capture of the create repository section inside a Github account.

STEP #3: Create Branches inside your REPO



The screenshot shows the GitHub interface for a repository named 'CasitaEscuela / power-bi-version-control-demo2'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A search bar is located on the right. The main section is titled 'Branches' and features a 'New branch' button. Below this, there are tabs for 'Overview', 'Yours', 'Active', 'Stale', and 'All'. A search bar for branches is also present. The 'Default' section shows a table with one branch, 'main', which is the default branch and was updated 1 hour ago. The 'Your branches' section shows a table with two branches, 'version2' and 'version1', both updated 1 minute ago. The 'Active branches' section also shows a table with 'version2' and 'version1' branches, both updated 1 minute ago.

Branches New branch

Overview Yours Active Stale All

Search branches...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	1 hour ago				Default

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
version2	1 minute ago		0	0	
version1	1 minute ago		0	0	


Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
version2	1 minute ago		0	0	
version1	1 minute ago		0	0	

A capture of the Branches section inside a Github repository.

STEP #4: Clone a Git REPO from Github (The CLI way...)

```
C:\Users\agonz615>  
git clone https://github.com/your-github-username/your-repo-name.git
```

 Command Prompt

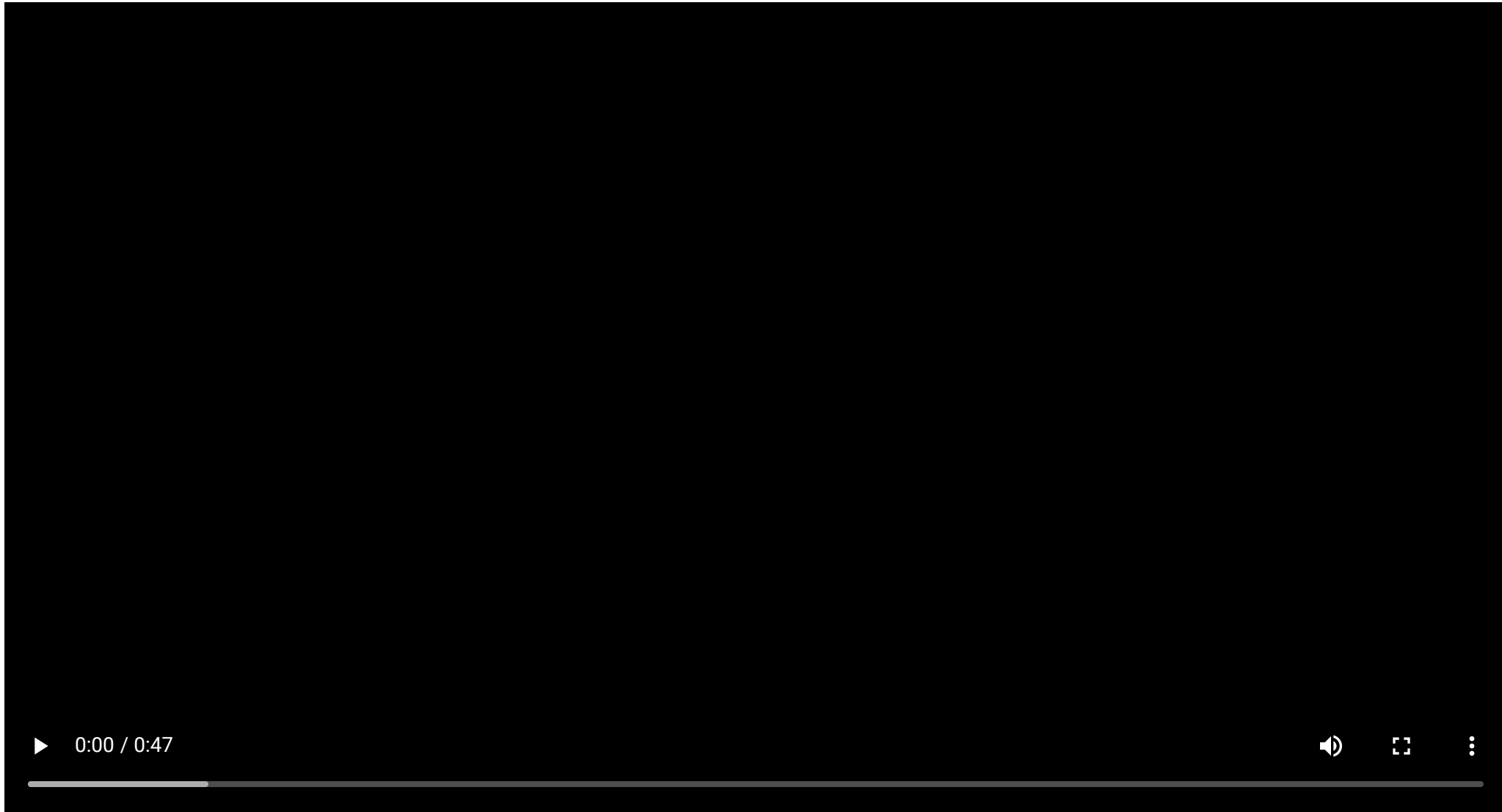
```
C:\Users\agonz615>git clone https://github.com/CasitaEscuela/power-bi-version-control-demo1.git  
Cloning into 'power-bi-version-control-demo1'...  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (4/4), 12.85 KiB | 67.00 KiB/s, done.  
  
C:\Users\agonz615>
```

A (Windows) command prompt showing the git clone command.

STEP #4: Clone a Git REPO from Github (The VSCode way...)



STEP #5: Change your branch A.K.A checkout (The VSCode way...)



LET'S DO THIS 

Now make changes and keep saving using the **.pbix** file format.